User's manual of the B_exclusive software package

B. Hoeneisen

DØ note 4772,
6 April 2005

Abstract

This note is the user's manual of the B_exclusive software package (version 6-APR-05). This software reads DØ data in tmb_tree format, and searches for exclusive particles of interest for B-physics analysis.

1 Introduction

The software package B_exclusive reads DØ data in tmb_tree format, and searches for exclusive particles of interest for B-physics analysis. Examples are:
$J/\psi \to \mu^+\mu^-$,
Ks$\to \pi^+\pi^-$,
$\Lambda \to p^+\pi^-$,
$\phi(1020) \to K^+ K^-$,
$K^*(892)^+ \to K^0\pi^+$,
$K^*(892)^0 \to K^+\pi^-$,
$\rho(770) \to \pi^+\pi^-$,
$D(1865)^+ \to \phi(1020)\pi^+$,
$D(1865)^0 \to K^-\pi^+$,
$D(1865)^+ \to \bar{K}^*(892)^0\pi^+$,
$D(1865)^0 \to K^*(892)^-\pi^+$,
$D(1865)^0 \to \bar{K}^0\pi^+\pi^-$,
$D(1865)^+ \to \bar{K}^0\pi^+$
$D^*(2010)^+ \to D^0\pi^+$,
Bd $\to J/\psi\ K^0$,
Bs $\to J/\psi\ \phi(1020)$,

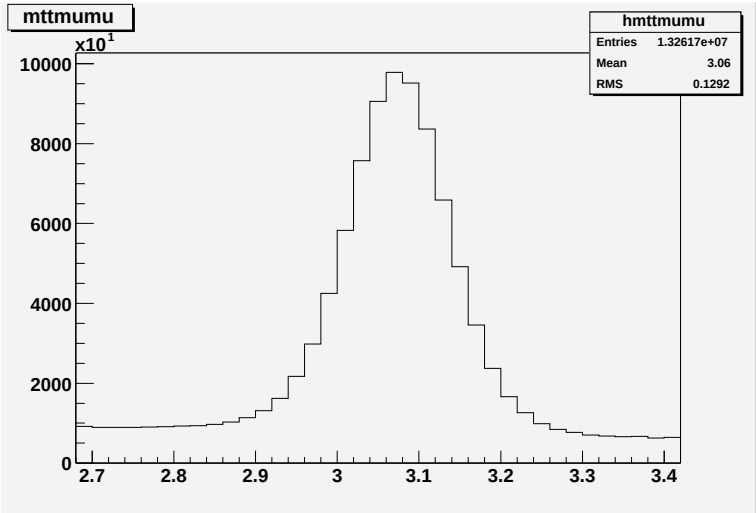

Figure 1: $J/\psi \rightarrow \mu^+\mu^-$ using function Track_Track.

Bd $\rightarrow J/\psi$ $K^*(892)^0$,
$B(5279)^+ \rightarrow J/\psi$ K^+,
$B(5279)^+ \rightarrow J/\psi$ $K^*(892)^+$,
$B(5279)^+ \rightarrow \bar{D}^0\pi^+$,
$\Lambda_c(2285)^+ \rightarrow \Lambda\pi^+$,
$\Lambda_b(5624) \rightarrow J/\psi\Lambda$.

A sample of invariant mass histograms are given in Figures 1 to 9. These histograms correspond to the J/ψ skim of Rounds 1 through 5 of data (known to the B-physics group). Note that the cuts have not been optimized for any specific analysis.

Other, more developed, packages that do the same are Ariel's (*et. al.*) d0root (also reads data in tmb_tree format), and Guennadi's (*et. al.*) BANA (which reads data in bana format).

2 Installation

How to install (this worked on clued0 on 6-APR-05):
mkdir new_p16.05.02 (or whatever name you choose for the directory where the code will be installed)
cd new_p16.05.02
setup D0RunII p16.05.02 (or whatever release you choose)
setup d0cvs

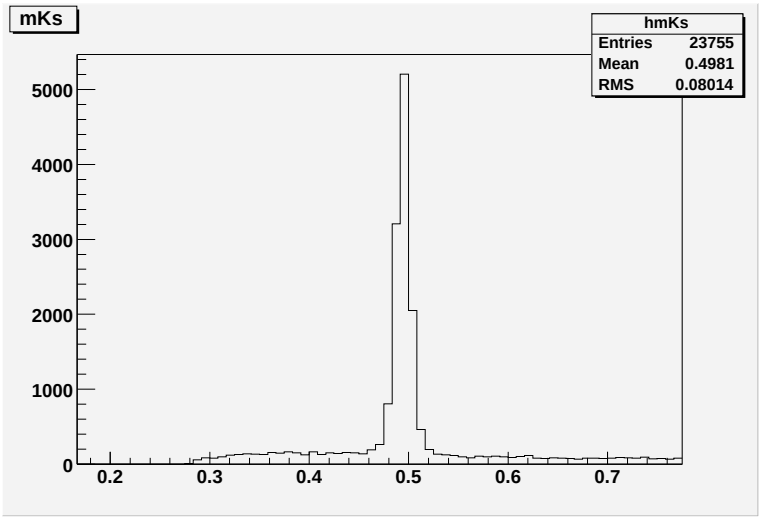

Figure 2: Ks $\rightarrow \pi^+\pi^-$ using function find_track_pairs.

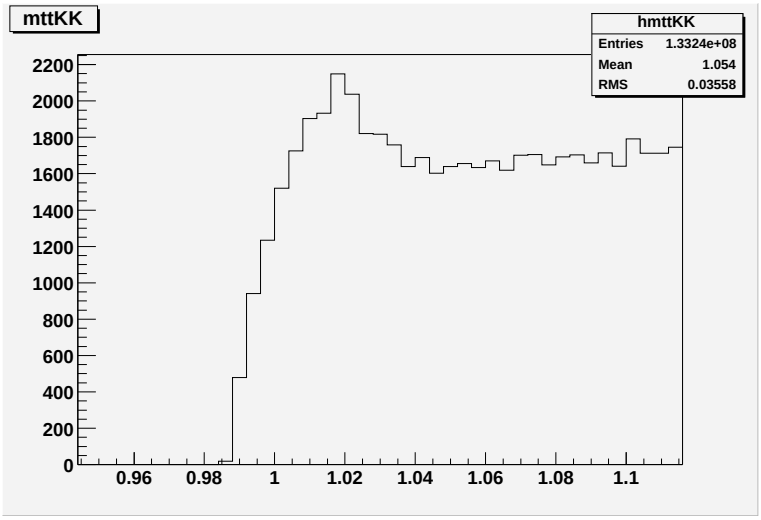

Figure 3: $\phi(1020) \rightarrow \mathrm{K}^+\ \mathrm{K}^-$ using function Track_Track.

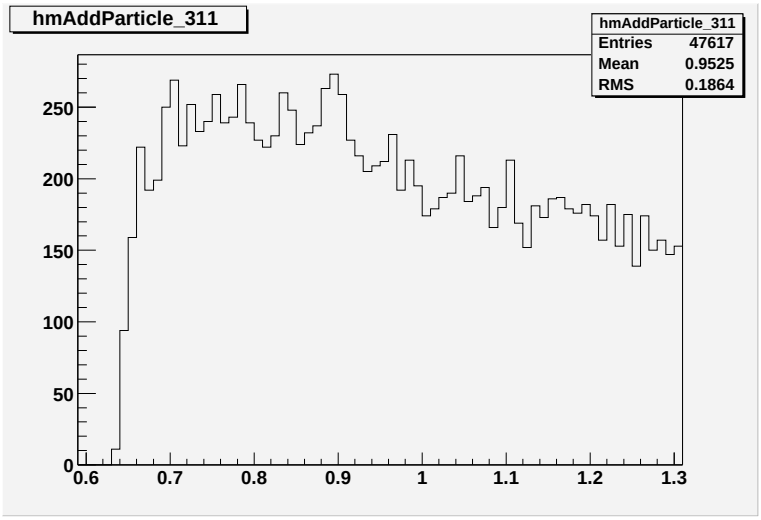

Figure 4: $K^*(892)^+ \to K^0\pi^+$ using function AddParticle.

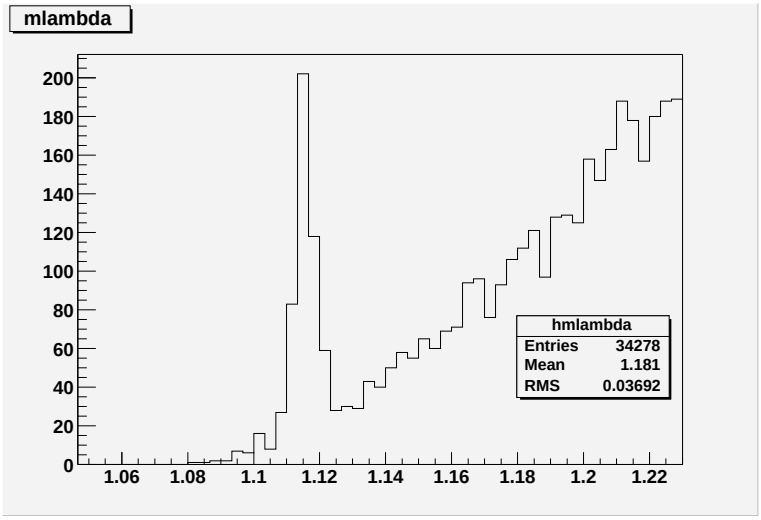

Figure 5: $\Lambda \to p^+\pi^-$ using function find_track_pairs.

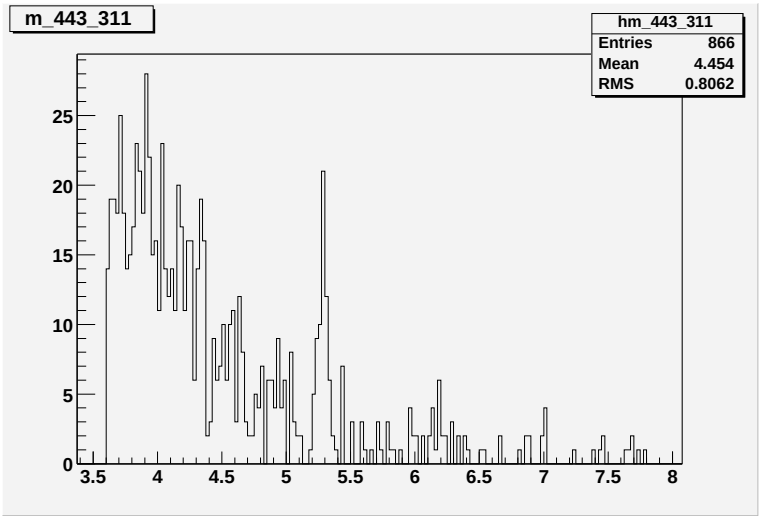

Figure 6: Bd $\rightarrow$J/ψ K^0 using function Particle_Particle.

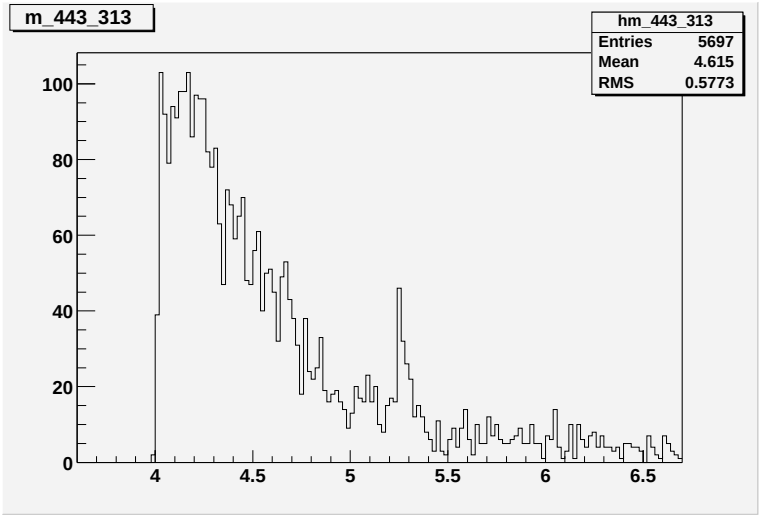

Figure 7: Bd $\rightarrow$J/ψ $K^*(892)^0$ using function Particle_Particle.

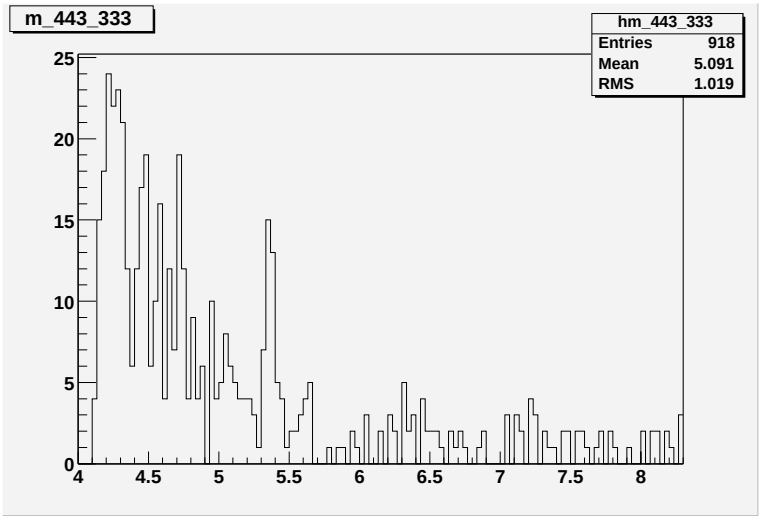

Figure 8: Bs $\rightarrow$J/ψ $\phi(1020)$ using function Particle_Particle.

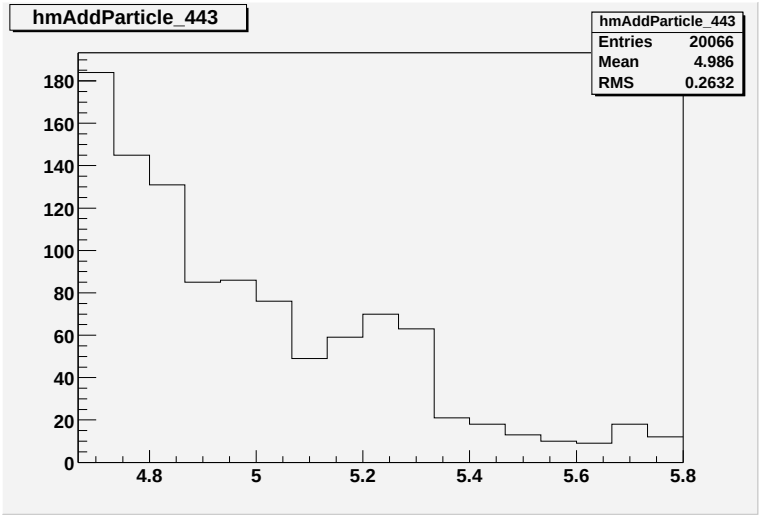

Figure 9: $B(5279)^+ \rightarrow J/\psi\ K^+$ using function AddParticle.

setup root v4_00_02b -q GCC_3_1:exception:opt:thread (at least for linux on clued0)
d0setwa
kinit -f and kerberos password
cvs checkout tau_cand
cd tau_cand
Go to the directory where Select.C and Select.h are, probably:
cd macros
MakeTMBTreeClasses_so.C should be there (but sometimes it is not). If it is not, you should get it from /d0dist/dist/releases/p16.05.02/tmb_tree/macros. [1]

Rename Select.C and Select.h since they will be replaced later:
mv Select.C Select.C_tau_cand
mv Select.h Select.h_tau_cand

Copy B_exclusive.tar_060405 to your area:
cp ~bruce1/B_exclusive_d0note/B_exclusive.tar_060405 . [2]
tar xvf B_exclusive.tar_060405

Now compare Select.C with Select.C_tau_cand and Select.h with Select.h_tau_cand and see if any modifications need to be done to Select.C and Select.h for your particular release (if other than p16.05.02).

Edit making_chain.C (which creates the object TTree MuTrk) with your list of tmb_tree's. An example provided is making_new_chain_rounds_1_6_Jpsi.C_3. The format is:

```
{
TChain MuTrk("TMBTree");
MuTrk.Add("/prj_root/1152/ckm/rickv/2MU/round1/filtered772093_last.root");
MuTrk.Add("/prj_root/1152/ckm/rickv/2MU/round1/filtered772095_1.root");
}
```

Edit run.C to point to the making_chain.C you wish to analyze.
Finally edit rsubmit4.sh to point to your directories.
The output text will land in job.out and the output histograms will land in histos.root.

How to run in batch mode:
cluesow -l cput=35:00:00 -l mem=500M rsubmit4.sh

[1] I have included in B_exclusive.tar the missing file MakeTMBTreeClasses_so.C for release p16.05.02 with the name MakeTMBTreeClasses_so.C_p160502. So, if you do not have MakeTMBTreeClasses_so.C, and you are using p16.05.02, then you can cp MakeTMBTreeClasses_so.C_p160502 MakeTMBTreeClasses_so.C.

[2] A copy is also kept in /work/naia-clued0/bruce1/B_exclusive/tau_cand.

Short form of interactive running:

```
root -b
.x run.C
```

Long form of interactive running:
Setup your window if you have not done so before:

```
cd (path)/new_p16.05.02
setup D0RunII p16.05.02
setup root v4_00_02b -q GCC_3_1:exception:opt:thread
d0setwa
cd tau_cand/macros
```

Now run interactively:

```
root -b
.x MakeTMBTreeClasses_so.C
.L Select.C++
```

For reading one tmb_tree file:

```
Select mu("(path).root");
```

Instead, for reading a list of tmb_tree files:

```
.x (path)/making_chain.C
Select mu(& MuTrk);
// mu._nwrite=100; // to write out 100 events
// mu.write_tree_to("tmb_tree_skim.root");
```

The preceeding 2 commented commands for skimming are optional and require you first edit B_exclusive_loop.C accordingly.

```
mu.Loop();
mu.write_histos("histos.root"); // Output histograms will be in histos.root
.q
```

3 Overview

The tau_cand package in d0cvs provides a nearly empty skeleton into which users can insert their analysis code. B_exclusive inserts code into Select.h and Select.C of the tau_cand package. The driving program of B_exclusive is B_exclusive_loop.C described in Section 4. The usage of functions can be found in B_exclusive_loop.C. Most functions of B_exclusive are collected in the file B_exclusive_functions.C. The most important functions are Track_Track, find_jpsi_ks, AddParticle and Particle_Particle described in Sections 5 through 8. Understanding these functions it should be relatively easy to edit B_exclusive_loop.C for any particular analysis. The B_exclusive package also

includes classes of the package find_jpsi_ks, which are used to search for the long-lived particles Ks and lambda. All files are described in Section 9.

4 B_exclusive_loop.C

This is the driving program of the B_exclusive package. It should be edited to meet the specific requirements of the user. The usage of functions can be found here.

Version 6-APR-05 of B_exclusive_loop.C does the following main chores:

- Initiates the list of particles lp of this event.
- Obtains the beam position (xbeam, ybeam) at $z = 0$ (using function GetBeam in GetPV.h) and makes histograms.
- Selects events of interest by trigger name.
- Marks repeated muons to avoid their use (the MarkRepeatedMuons() function sets the flag mark[i] to 1 for best among repeated muons).
- The function Zmuon() obtains zmuon of highest p_T muon and sets pxjet, pyjet, pzjet, zjet to muon momentum and z (this is used as a seed for GetJetP).
- Obtains the beam position (xbeam, ybeam) at $z =$ zmuon.
- Using function GetJetP obtains pxjet, pyjet, pzjet, zjet of jet corresponding to the highest p_T muon.
- Uses function Track_Track to search for $J/\psi \to \mu^+\mu^-$, and adds them to the list of particles lp.
- Uses function Track_Track to search for $K^*(892)^0 \to K^+\pi^-$, $D(1865)^0 \to K^-\pi^+$, and $\phi(1020) \to K^+ K^-$ and adds them to the list of particles lp.
- Using function find_track_pairs (in Find_Track_Pairs.cpp) searches for Ks $\to \pi^+\pi^-$ and $\Lambda \to p^+\pi^-$, and adds them to the list of particles lp.
- Using function AddParticle or Add2Particle searches for $K^*(892)^+ \to K^0\pi^+$, $D^+ \to \bar{K}^0\pi^+$, $D^+ \to \bar{K}^0K^+$, $D(1865)^0 \to \bar{K}^0\pi^+\pi^-$, $D^*(2010)^+ \to D^0\pi^+$, $\Lambda_c(2285)^+ \to \Lambda\pi^+$, and $D(1865)^+ \to \phi(1020)\pi^+$.
- Using the function Particle_Particle searches for $D^+ \to K^*(892)^+\bar{K}^0$ and $D^+ \to K^*(892)^+\bar{K}^*(892)^0$.

- Then the search is narrowed to events with a reconstructed J/ψ.
- Using function GetJetP obtains new pxjet, pyjet, pzjet, zjet of jet of the J/ψ.
- Uses function Track_Track to search for $\rho(770) \rightarrow \pi^+\pi^-$, $K^*(892)^0 \rightarrow K^+\pi^-$, and $\phi(1020) \rightarrow K^+ K^-$.
- Uses function AddParticle to search for $B^+(5279) \rightarrow J/\psi\ K^+$.
- Uses function Particle_Particle to search for Bd $\rightarrow$J/ψ Ks, Bs $\rightarrow$J/ψ $\phi(1020)$, $\Lambda_b(5624) \rightarrow$ J/$\psi\Lambda$, Bd $\rightarrow$J/ψ $K^*(892)^0$, and $B^+ \rightarrow$J/ψ $K^*(892)^+$.

5 Function Track_Track

The function Track_Track returns the invariant mass of two tracks and adds the parent to the list of particles of the event (LParticle lp) if within mass windows (or successfull mass-constrained-fit). If the two tracks fail to pass cuts then an error code is returned (which is negative). The declaration of the function is:

double Select::Track_Track(int itrk1, int itrk2, double m1, double m2, int opcharge, double xvertex, double yvertex, double zvertex, double mind, double maxd, double pTmintrk, double pTminparent, double decaylengthsig, double mc)

itrk1 and itrk2 are the track numbers (as in the piece of code below). Tracks are required to pass cuts defined in function PassCutsT (to be optimized by the user for his or her particular analysis), and are required to not correspond to any used track of particles in lp. m1 and m2 are the masses assigned to these tracks (in GeV). If opcharge == 1 then the charges of the two tracks are required to be opposite. xvertex, yvertex, zvertex are the coordinates of the primary vertex with respect to which we calculate the impact parameter and decay length of the parent (projected onto the momentum of the parent) in the transverse plane. The impact parameter in the transverse plane is required to be less than 1.0 millimeters (this cut should be optimized for a particular analysis). The decay length in the transverse plane is required to be in the range mind to maxd. The transverse momentum (in GeV) of the track is required to be greater than pTmintrk, and the transverse momentum of the parent is required to be greater than pTminparent. If mc $>$ 0.0, a mass-constrained-fit is performed to mass mc (the user should optimize the parameter epsilonmax which defines the range of a good mass-constrained-fit).

The function Track_Track requires that (decay length)/(decay length error) > decaylengthsig (set decaylengthsig = -9999.0 if you do not want this cut). A piece of code using Track_Track is

```
for(int i=0; i < (int)fTrks->GetLast(); i++)
{
  for(int j=i+1; j < (int)fTrks->GetLast()+1; j++)
  {
    if(isSameJetAsSeed(i, DR) && isSameJetAsSeed(j, DR))
    {
      m = Track_Track(i, j, m1, m2, opcharge, xbeam, ybeam, zmuon,
        mind, maxd, pTmintrk, pTminparent, decaylengthsig, mc);
      hmttmumu_trk->Fill(m);
    } // end isSameJetAsSeed
  } // end j
} // end i
```

6 Function find_track_pairs

The function Find_Track_Pairs::find_track_pairs is similar to function Track_Track, except that it uses the full helical parametrization of the tracks, which is necessary for long-lived particles such as Ks and lambda. The class Find_Track_Pairs is part of the package find_jpsi_ks in d0cvs. The class Find_Track_Pairs is constructed with an RCP file corresponding to the particle that is sought. The declaration of the function is:

void Find_Track_Pairs::find_track_pairs(LTrackFound_xyz &ltfxyz, LDecayPair &ldp, float Bmag)

Given the magnetic field intensity at the center of the solenoid Bmag (in Tesla), and a list of tracks LTrackFound_xyz &ltfxyz (in the format used in find_jpsi_ks), the function find_particle_pairs returns a list of parents LDecayPair &ldp (in the format used in find_jpsi_ks) that meet the requirements of the RCP file.

7 Function AddParticle

Function AddParticle adds a track to a particle in lp, and adds the resulting parent to the list of particles of the event (LParticle lp) if within mass win-

dows (or successfull mass-constrained-fit). The declaration of the function is:

int Select::AddParticle(int ID, int charge, double mass,
double decaylength_particle_min, double decaylength_particle_max, double decaylength_parent_min, double decaylength_parent_max, double mass_min, double mass_max, double pTmin, double mc)

Function AddParticle first loops over the list of particles of the event (LParticle lp) to find particles of type ID (as in the Monte Carlo particle numbering scheme of the Particle Data Group). Then it loops over tracks and builds parents. The tracks are required to pass cuts defined in function PassCutsT, not be a used track of any of the particles in lp, have charge charge, and be within DR of the jet (the user should change the default DR = 20.0 if he or she wants to apply this constraint). The track is asumed to have mass mass. The track is required to be within deltaR = 1.0 of the particle in eta-phi space (the user should optimize this cut). The z-coordinates of the particle and track at their point of crossing in the transverse plane are required to be within 10 millimeters (the user should optimize this cut). The decay length of the particle (projected onto the momentum of the particle) in the transverse plane is required to be between decaylength_particle_min and decaylength_particle_max. The transverse momentum of the particle is required to be greater than pTmin. The decay length of the parent (projected onto the momentum of the parent) in the transverse plane is required to be between decaylength_parent_min and decaylength_parent_max. The impact parameter in the transverse plane of the parent with respect to the beam is required to be less than 0.3 millimeters or (decaylength_parent $>$ 2.0 && cos2 $>$ 0.9) (this cut should be optimized by the user). mass_min and mass_max are currently not used. If mc $>$ 0.0, a mass-constrained-fit is performed to mass mc (the user should optimize the parameter epsilonmax which defines the range of a good mass-constrained-fit). Function AddParticle currently returns zero.

The function Add2Particles is similar to the function AddParticle except that two tracks (instead of one) are added to a particle.

8 Function Particle_Particle

The function Particle_Particle sums particles in the list lp with ID1 and ID2 and plots the invariant mass of the parent. The declaration of the function is:

bool Select::Particle_Particle(int ID1, int ID2, int charge, double decay_length_min,

double decay_length_max, double pTmin, double decaylength1min, double decaylength1max, double decaylength2min, double decaylength2max)

The charge of the parent is required to be charge. The z-coordinates of the particles at their point of crossing in the transverse plane are required to be within 10 millimeters (this cut should be optimized by the user). The decay length of the parent with respect to the beam (projected onto the momentum of the parent) in the transverse plane is required to be in the range decay_length_min and decay_length_max, and similarly for particle 1 and particle 2 with respect to the parent's decay point. The impact parameter of the parent in the transverse plane (with respect to the beam) is required to be less than 1 millimeter (the user should optimize this). The transverse momentum of the parent is required to be greater than pTmin. Currently the function Particle_Particle returns true.

9 List of files

- Select.h and Select.C: Part of the tau_cand package. We include most of the software of the B_exclusive package into these two files.
- B_exclusive.h: included into Select.h, defines variables, histograms, and functions used by the B_exclusive package.
- B_exclusive_init.C: included into Select.C. Initializes all histograms.
- B_exclusive_loop.C: included into Select.C. This is the driving program of the B_exclusive package. The usage of functions can be found here.
- B_exclusive_write_histos.C: included into Select.C. Writes all histograms after the end of the loop.
- B_exclusive_endofloop.C: included into Select.C. For the moment it is empty.
- B_exclusive_functions.C: included into Select.C. Collects most functions used in B_exclusive_loop.C.
- GetPV.h and GetPV.C: Contains function GetBeam that, given the run number and z, returns the beam position xbeam, ybeam. Author ?
- making_new_chain_rounds_1_6_Jpsi.C_3: Example of a list of files containing the tmb_tree's.

- run.C: List of directives to the root program. Can be run manually following these commands, or in batch mode using rsubmit4.sh. User must edit run.C to point to correct making_chain.C file.
- rsubmit4.sh: Used to submit batch job with the command: cluesow -l cput=12:00:00 -l mem=500M rsubmit4.sh. User must edit rsubmit4.sh to point to correct files.
- Find_Track_Pairs.hpp and Find_Track_Pairs.cpp: Contains the function find_track_pairs used to reconstruct long-lived particles such as Ks and Λ that decay into two tracks.
- TrackFound_xyz.hpp and TrackFound_xyz.cpp: Object containing information of one track. Used by the function find_track_pairs to search long-lived particles that decay into two tracks.
- LTrackFound_xyz.hpp and LTrackFound_xyz.cpp: List of TrackFound_xyz's.
- DecayPair.hpp and DecayPair.cpp: Object that contains a particle that decays into two tracks. Used by the function find_track_pairs in Find_Track_Pairs.cpp.
- LDecayPair.hpp and LDecayPair.cpp: List of DecayPair's. Used to search long-lived particles such as Ks and Λ.
- Find_Track_Pairs_Jpsi.RCP, Find_Track_Pairs_KK.RCP, Find_Track_Pairs_Ks.RCP, Find_Track_Pairs_lambda2.RCP, Find_Track_Pairs_lambda.RCP, and Find_Track_Pairs_pipi.RCP. RCP files used by function find_track_pairs to search particles that decay into two tracks.
- gen_circle_intercept.cpp: Function used by find_track_pairs.
- get_alpha.cpp: Function used by find_track_pairs.
- Particle.hpp: Object that describes a particle and its decay tracks.
- LParticle.hpp: List of Particle's.

References

[1] "Documentation for the 'Find_Jpsi_Ks' software that searches tracks and track pairs using the Fiber Tracker", B. Hoeneisen, DØ note 3842 (2001).